IN THE UNITED STATES DISTRICT COURT
EASTERN DISTRICT OF VIRGINIA
ALEXANDRIA DIVISION

BMG RIGHTS MANAGEMENT (US) LLC, and ROUND HILL MUSIC LP, Plaintiffs, v. COX ENTERPRISES, INC., COX COMMUNICATIONS, INC., and COXCOM, LLC, Defendants.	)	Case No. 1:14-cv-1611 (LO/JFA)

DECLARATION OF CHRISTOPHER T. RUCINSKI IN SUPPORT OF COX'S MOTION FOR EVIDENTIARY SANCTIONS BASED ON SPOLIATION OF RIGHTSCORP SOURCE CODE

REDACTED NON-CONFIDENTIAL VERSION

I, Christopher T. Rucinski, submit this declaration pursuant to 28 U.S.C. § 1746.

1. I am a computer scientist working at Elysium Digital, L.L.C. ("Elysium Digital"), a technical litigation consulting company in Boston, MA. Counsel for Cox Communications, Inc. and CoxCom, LLC (collectively, "Cox") retained me to conduct expert analysis and provide expert testimony in this case. In this declaration specifically, I have been asked to address a question from the Court regarding "whether the heart of the [Rightscorp software] system was the same based on what you have in 2013 and the version that was produced for July of 2015." I refer to certain source code files produced by Rightscorp, Inc. that were supposedly in operation in 2013 as the "Produced Historical Rightscorp Source Code" and Rightscorp's systems as they existed for some time in 2013 as the "Historical Rightscorp Systems." In July 2015, Rightscorp produced source code files for its current software systems, which I refer to as the "Current Rightscorp Systems." In this declaration I will refer to the Historical Rightscorp Systems and the Current Rightscorp Systems collectively as the "Rightscorp Systems." I conclude that the Produced Historical Rightscorp Source Code is not the "heart" of the Rightscorp Systems and provides very little information about how the Historical Rightscorp Systems functioned.

2. The facts in this declaration rest upon my personal knowledge and analysis of the source code that Plaintiffs' agent Rightscorp produced in this lawsuit. I also rely on certain testimony from the depositions of Rightscorp personnel and Plaintiffs' technical expert Barbara Frederiksen-Cross. I personally attended three of those depositions, including the July 29, 2015 Rule 30(b)(6) deposition of Greg Boswell of Rightscorp, who wrote the Rightscorp software; the July 29, 2015 Rule 30(b)(6) deposition of Robert Steele of Rightscorp, who created the specifications for the Rightscorp software and supervised its development; and the August 12, 2015 expert deposition of Ms. Frederiksen-Cross.

3. I have submitted two expert reports in this matter. My first expert report ("Rucinski Rebuttal") was submitted on July 10, 2015. I understand that a complete copy of Rucinski Rebuttal is in the record in this case as Doc. 271, Ex. 2. My second supplemental expert report ("Rucinski Supplemental") was submitted on July 31, 2015. I understand that a complete copy of Rucinski Supplemental is in the record in this case as Doc. 256.1, Ex. 28. I also submitted a prior declaration in this matter, "Declaration of Christopher T. Rucinski in Support of Defendants' Motion for Summary Judgment" ("Rucinski SJ Declaration"), on September 21, 2015. I understand that a complete copy of Rucinski SJ Declaration is in the record in this case as Doc. 313. The facts and opinions I express in Rucinski Rebuttal, Rucinski Supplemental, and Rucinski SJ Declaration are consistent with the facts and opinions I express here.

4. As an initial matter, it is misleading to talk about the "heart" of a software system when discussing its operation because most software systems of any nontrivial complexity comprise a multitude of interdependent parts. The removal of, addition to, or change of any single one of those parts, no matter how small, can dramatically affect the operation of the system as a whole. Even small errors in a few lines of source code can have dramatic effects on the operation of the corresponding system as a whole. (For example, the omission of a single hyphen character in source code resulted in the explosion of NASA's Mariner 1 spacecraft.)

5. In this case, we have seen how a single small file can dramatically affect the operation of a software system. In the Current Rightscorp Systems, the 10% bitfield threshold is implemented in a file on the periphery of the source code; but for the inclusion of that peripheral file, a 100% bitfield threshold would be used by those systems instead. That small file is `FullFilesBeing.txt`,[1] and it is only 22 lines long when printed. I discuss the operation and

[1] In this declaration, terms in Courier font refer to specific files in the Rightscorp software.

production of the code related to the 10% bitfield threshold in Rucinski SJ Declaration at ¶19.[2] There are many other processes or routines that run at the periphery of the main source code modules but that can have a huge impact on how the Rightscorp Systems function overall.

6. Even if it were possible to identify a body of source code that is the "heart" of a software system, if that software system relates to identifying and verifying data sources, collecting data, processing that data, and outputting data, it is impossible to know the quality of the outputted data without knowing the quality of the collected data. Even if the data processing methods present in the software system are sound, consistently incorrect data inputs to the software system will still result in consistently incorrect data outputs from the system. This is often referred to as the "garbage in, garbage out" principle.

7. I discuss the Produced Historical Rightscorp Source Code at some length in Rucinski Supplemental at ¶¶29-49. The Produced Historical Rightscorp Source Code does include source code files such as `Test5.java`, `TorrentRequest.jsp`, and `Detected.jsp`, which relate to recording observations about the bitfields of peers in a BitTorrent swarm for particular .torrent files. However, the Produced Historical Rightscorp Source Code does not indicate the origin of the .torrent files that the Historical Rightscorp Systems used, or how or whether the Historical Rightscorp Systems compared those .torrent files to a list of works at issue in this lawsuit.[3] The Produced Historical Rightscorp Source Code also does not indicate how the Historical Rightscorp Systems verified sound recording files from torrent payloads, if in fact those systems did so.[4]

---

[2] *See also* Rucinski Rebuttal ¶¶24-27, 43.

[3] *See* Rucinski Supplemental at ¶¶31-34.

[4] *See* Rucinski Supplemental at ¶¶35-36.

8. The Produced Historical Rightscorp Source Code does not include any substance regarding how the Historical Rightscorp Systems calculated the appropriate number of notifications of claimed infringement to send.[5] The Produced Historical Rightscorp Source Code also does not include any substance relating to the actual generation of notifications of claimed infringement, and it does not include any substance relating to the identification of IP address and port combinations that Rightscorp might identify as "repeat infringers".[6]

9. To even consider the "heart" of the Historical Rightscorp System, misleading though that concept inherently may be, I really must know how the entire Historical Rightscorp Systems functioned. Because I have only limited portions of the source code for the Historical Rightscorp Systems, I am forced to make many critical assumptions about the operation of those systems. Because the Produced Historical Rightscorp Source Code omits source code modules necessary for the operation of the Historical Rightscorp Systems, I *assume* exclusively for the purposes of considering what might be the "heart" of the Rightscorp Systems, that the many missing source code modules in the Historical Rightscorp Systems operated in essentially the same way that corresponding modules operate in the Current Rightscorp Systems, except where the existing evidence indicates the contrary.

10. The above assumption is highly favorable to Rightscorp because not only do I not know how certain aspects of the Historical Rightscorp Systems operated, I do not even know *what I do not know* about how certain aspects of the Historical Rightscorp Systems operated. It could be that source code modules in the Historical Rightscorp Systems operated substantially differently from the corresponding modules in the Current Rightscorp Systems. It also could be

[5] *See* Rucinski Supplemental at ¶38.

[6] *See* Rucinski Supplemental at ¶¶39-40.

that there were source code modules that were important to the operation of the Historical Rightscorp Systems that were later removed; I would not be aware of the existence of such modules. Barbara Frederiksen-Cross never analyzed the modules missing from the Produced Historical Rightscorp Source Code because Rightscorp did not give them to her, and Greg Boswell has only his memory to rely upon and could not recall numerous details about the operation of the Historical Rightscorp Systems.

11. As reflected in the Reply Report of Barbara Frederiksen-Cross ("Frederiksen-Cross Reply"), submitted on July 24, 2015 and corrected on July 27, 2015, the Produced Historical Rightscorp Source Code does not include any source code related to important functionalities of the current Rightscorp system, such as ingesting copyrighted works, finding and filtering .torrent files on the Internet, and verifying the contents of .torrent files, because in 2013, those processes were performed manually.[7] The opening Report of Barbara Frederiksen-Cross ("Frederiksen-Cross Opening"), submitted on June 19, 2015, indicates that, [redacted][8] Audible Magic and AcoustID are fingerprinting processes implemented to correct one source of errors in the Current Rightscorp Systems.[9] I have no way of determining the accuracy of the above dates related to the implementation of this important process, but a fingerprinting process apparently was not in place in June 2013 when Ms. Frederiksen-Cross reviewed portions of the source code for the Historical Rightscorp Systems.

12. As I discuss in my July 10 report, the key elements of the Current Rightscorp Systems are source code modules that relate to the following four key operations: (1) attempting

[7] *See* Doc. 313, Ex. 13 (Frederiksen-Cross Reply) at ¶63.

[8] *See* Doc. 313, Ex. 14 (Frederiksen-Cross Opening) at ¶71.

[9] *See* Rucinski SJ Declaration at ¶28.

to verify that torrent payloads corresponding to identified .torrent files contain songs related to copyrighted works; (2) collecting data about the bitfields of peers in BitTorrent swarms associated with those .torrent files; (3) analyzing that data to determine which notifications of claimed infringement to send; and (4) sending notifications of claimed infringement based on that analysis. As I explain below, only one of these four key modules is present in the Produced Historical Rightscorp Source Code.

13. Attempting to verify that torrent payloads corresponding to identified .torrent files contain songs related to copyrighted works is the first core functionality of the Rightscorp Systems. In the Current Rightscorp Systems, this functionality is implemented in at least `SampleIt3.java`, `SampleRequest.jsp`, `SFT.jsp`, `MusicDownload1.java`, and `MusicDownloadPDCleanUp1.java`.[10] The Produced Historical Rightscorp Source Code does not include any source code files with those names or any source code files that implement similar functionality presumably because, as discussed above, in 2013 some incarnation of this process was performed manually rather than in an automated fashion.

14. Collecting data about the bitfields of peers in BitTorrent swarms associated with .torrent files is the second core functionality of the Rightscorp Systems. In the Current Rightscorp Systems, this functionality is implemented in at least `Test5.java`, `TorrentRequest.jsp`, `Detected.jsp`, `FullFilesBeing.txt`, and `FullFileFix.txt`.[11] The Produced Historical Rightscorp Source Code includes versions of `Test5.java`, `TorrentRequest.jsp`, and `Detected.jsp`. However, I cannot know whether there were other source code files in the Historical Rightscorp Systems that are relevant

[10] *See* Rucinski Rebuttal Sec. III(E).

[11] *See* Rucinski Rebuttal Sec. III(F).

to this functionality, such as files similar to `FullFilesBeing.txt` and `FullFileFix.txt`, which exist in the Current Rightscorp Systems and dramatically impact the output of `Test5.java` and related files. Rightscorp produced no data from the *TorrentInfractions*[12] database table, which is where the output of `Test5.java` is stored. As a result, the actual (alleged) infraction data, including bitfield data, ultimately underlying all of the historical notifications of claimed infringement is also not available for my review.

15. Analyzing the bitfield data collected by `Test5.java` and related files is the third core functionality of the Current Rightscorp Systems. The manner in which collected data is processed, manipulated, and filtered is important to understand because this determines the conditions under which notifications of claimed infringement are generated. In the Current Rightscorp Systems, this functionality is implemented in at least `TorrentInfractionExpander.java` and `ExpanderToInfractions.java`.[13] The Produced Historical Rightscorp Source Code does not include any source code files with those names or any source code files that implement similar functionality.

16. Sending notifications of claimed infringement is the fourth core functionality of the Current Rightscorp Systems. Examining how the actual output of the Rightscorp Systems is generated is critically important to understanding those systems' efficacy and reliability. In the Current Rightscorp Systems, this functionality is implemented in at least `CEmail.java` and `CPost.html`.[14] The Produced Historical Rightscorp Source Code does not include any source code files with those names or any source code files that implement similar functionality.

---

[12] In addition to their standard usage regarding emphasis, I use italics in this declaration to identify the names of database tables used by the Rightscorp Systems.

[13] *See* Rucinski Rebuttal Sec. III(G).

[14] *See* Rucinski Rebuttal Sec. III(H).

17. The Produced Historical Rightscorp Source Code does include source code files such as `SampleIt2.java`, `DownloadRequest.jsp`, and `TFT.jsp`, which relate to downloading full file samples from individual peers in a BitTorrent swarm associated with a particular .torrent file.[15] But even assuming (favorably for Plaintiffs) that `SampleIt2.java` and related files in the Historical Rightscorp Systems operated in a manner similar to `SampleIt2.java` and related files in the Current Rightscorp Systems, those files do not map to any of the four key operations of the Rightscorp Systems that I identified above because the output of this module does not influence the ultimate output of the Rightscorp Systems as a whole, which is notifications of claimed infringement. In the Current Rightscorp Systems, `SampleIt2.java` and related files depend upon the operation of `TorrentInfractionExpander.java` and `ExpanderToInfractions.java`. Neither of those two files are included in the Produced Historical Rightscorp Source Code, and without that production, I cannot know how `SampleIt2.java` and related files interacted with the data from `Test5.java` and related files in the Historical Rightscorp Systems. In other words, the historical versions of `SampleIt2.java` and related files tell me very little, and nothing conclusive, about how the Historical Rightscorp Systems functioned. [redacted]

18. As I noted above, there are serious questions about whether it is ever possible to identify the "heart" of nontrivially complex software systems because the source code modules of those systems are so critically interdependent, and the functionality of such systems can be impacted by small changes. Moreover, to even discuss the "heart" of the Rightscorp Systems as

[15] *See* Rucinski Supplemental at ¶41.

compared to the Produced Historical Rightscorp Source Code, I must assume that the Current Rightscorp Systems and the Historical Rightscorp Systems functioned in essentially the same way where the existing evidence does not indicate the contrary; there is no reliable support for that assumption, and it is highly favorable to Rightscorp, which destroyed the best evidence to either confirm or refute that assumption. But if one assumes that the four key source code modules that I identified above and in my rebuttal report are the "heart" of the Current Rightscorp Systems, only one of those four modules is represented in the Produced Historical Rightscorp Source Code.

19. By the same token, it is impossible to assess whether the files in the Produced Historical Rightscorp Source Code are the "heart" of the Historical Rightscorp Systems because, based on these files alone, it is impossible to know how the Historical Rightscorp Systems actually functioned. Conspicuously lacking is the code related to analyzing observed (alleged) infringement data and generating notifications of claimed infringement based on that analysis. We can never know exactly how Rightscorp associated even one of the Plaintiffs' songs with a particular torrent payload because we do not have an evidentiary record of the relevant Rightscorp source code for any single date. The few full file samples that Rightscorp did produce in this case ([redacted]) demonstrate that the sampling process resulted in errors. To my knowledge, Rightscorp was unaware of those errors until I identified them. We also cannot know whether some peripheral process in the Historical Rightscorp Systems manipulated the output of `Test5.java` to change the bitfield threshold to something less than 100%, as `FullFilesBeing.txt` does in the Current Rightscorp Systems.

20. In summary, we know nothing from the Produced Historical Rightscorp Source Code about how the Historical Rightscorp Systems ingested copyrighted works, located .torrent files, expanded torrent-level infractions to song-level infractions (as

`TorrentInfractionExpander.java` and `ExpanderToInfractions.java` do in the Current Rightscorp Systems), and generated notifications of claimed infringement. In fact, we can never know how *any* particular historical notification of claimed infringement was generated because we do not have an evidentiary record of the relevant Rightscorp source code for any single date. The few files present in the Produced Historical Rightscorp Source Code certainly do not constitute the "heart" of the Rightscorp Systems and provides very little information about how the Historical Rightscorp Systems functioned.

I declare under the penalty of perjury that the foregoing is true and correct. Executed September 29, 2015.

Christopher T. Rucinski